***Cú pháp hook***

**Khung:**

**-------------------------------------------------**

**Const TenComponet =(){**

**//khoi tao state ( moi thuoc tinh 1 state)**

Const[tenState,setTenState]=useState(gia\_tri\_khoi\_tao\_state);

**//ham set state(dung useEffect) truyen vao 1 ham**

useEffect(() => {

        setTenState ((pre) => (

            thuộc tính cần định dạng lại

        ))

        return ()=>{

**// code thưc trước khi component unmount;**

        }

    }, [isLoading]);

**//return hiển thị dữ liệu**

return(

(chương trình html)

);

export default TenComponent;

**--------------------------------------------------------------------**

Vể Dependencies:

-k có tham số => hàm sẽ luôn dc chạy đến vô hạn

- mảng rỗng: chỉ chạy 1 lần

- có dependencies => khi giá trị thay đổi hàm sẽ dc chạy lại

Mỗi thuộc tính thì phải khai báo 1 state mới



Về các hàm ngoài:

    const handleIsLoading = ()=>{

        setIsLoading((prevState) => !prevState)

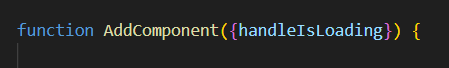
    } // chỉ thay đổi thuộc tính thôi chứ k dc cập nhật trên trang, muốn cập nhật phải có useEffect

Có thể truyền hàm vào file khác như truyền số vào hàm

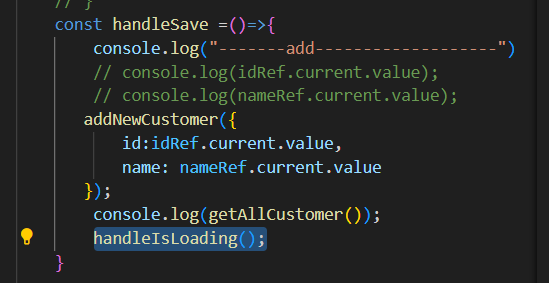


File add sử dụng như một đối số::

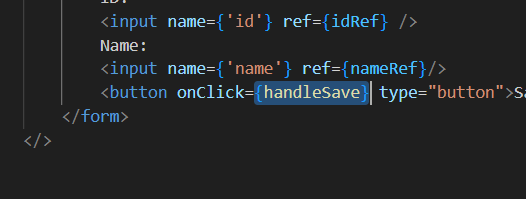
Nhận



Sử dụng

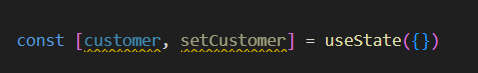


Truyền vào buttnon

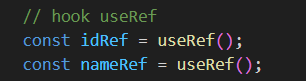


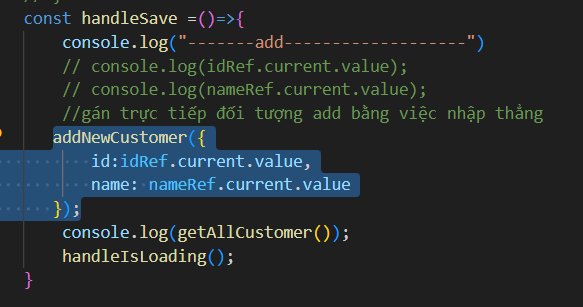
Lưu ý:

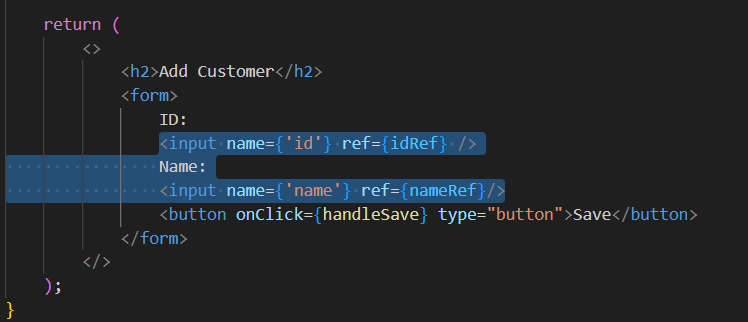
State cũng có thể là 1 đối tượng:=> khởi tạo {}



Cách nhập dữ liệu khác( sử dụng ref)







SO sánh giá trị cũ với giá trị mới dùng prevState luôn( nếu trong hàm chỉ có 1 giá trị thuộc tính

